Оглавление

[Общее 2](#_Toc164548214)

[Структурные диаграммы 3](#_Toc164548215)

[Диаграмма классов 3](#_Toc164548216)

[Межклассовые отношения 3](#_Toc164548217)

[Общее для межклассовых отношений 3](#_Toc164548218)

[Композиция 3](#_Toc164548219)

[Агрегация 4](#_Toc164548220)

[Ассоциация 4](#_Toc164548221)

[Использование 4](#_Toc164548222)

[Наследование\Генерализация 4](#_Toc164548223)

[Реализация 4](#_Toc164548224)

[Диаграмма объектов 5](#_Toc164548225)

[Диаграмма компонентов 6](#_Toc164548226)

[Диаграмма развертывания 7](#_Toc164548227)

[Диаграмма пакетов 9](#_Toc164548228)

[Диаграмма составной структуры 10](#_Toc164548229)

[Диаграмма профилей 11](#_Toc164548230)

[Диаграммы поведения 12](#_Toc164548231)

[Диаграмма прецедентов 12](#_Toc164548232)

[Диаграмма деятельности 13](#_Toc164548233)

[Диаграмма состояний 14](#_Toc164548234)

[Диаграмма взаимодействий 15](#_Toc164548235)

[Диаграмма последовательности 15](#_Toc164548236)

[Диаграмма коммуникаций 15](#_Toc164548237)

[Диаграмма обзора взаимодействий 15](#_Toc164548238)

[Временная диаграмма 15](#_Toc164548239)

Общее

Диаграмма классов UML иллюстрирует структуру системы, описывая классы, их атрибуты, методы и отношения между объектами.

Типы UML диаграмм:

1. Структурные диаграммы
   1. Диаграмма классов
   2. Диаграмма составной структуры
   3. Диаграмма развертывания
   4. Диаграмма пакетов
   5. Диаграмма профилей
   6. Диаграмма объектов
   7. Диаграмма компонентов
2. Диаграммы поведения
   1. Диаграмма деятельности
   2. Диаграмма прецедентов
   3. Диаграмма состояний
   4. Диаграмма взаимодействий
      1. Диаграмма последовательности
      2. Диаграмма коммуникации
      3. Диаграмма обзора взаимодействий
      4. Временная диаграмма

Структурные диаграммы

Диаграмма классов

Описывает классы их атрибуты методы и отношения между объектами

Центральная методика моделирования. Используется во всех программно-ориентированных методах. Описывает типы объектов в системе и виды статических отношений, которые существуют между объектами.

Виды перспектив:

1. Концептуальная
   1. Диаграмма интерпретируется как описание вещей в реальном мире
   2. Считается независимой от языка
2. Спецификационная
   1. Диаграмма интерпретируется как описание абстракций
   2. Без привязки к конкретной реализации на с привязкой к языку
3. Имплементационная
   1. Интерпретируется как описание реализаций на коде (имеет элементы из ЯПа (прим типы данных, функции …))

Межклассовые отношения

Делится все на:

1. Ассоциация.
2. Наследование\Генерализация (такая же функция, как и в ооп)
3. Реализация (подразумевается отношение интерфейса и объектов)
4. Зависимость (Объект одного класса использует объект другого класса в своем методе.
5. Агрегация (один класс является частью другого. Подклассы не уничтожаются, когда уничтожается объединяющий класс)
6. Композиция (один класс является частью другого. Подклассы уничтожаются, когда уничтожается объединяющий класс)

### Общее для межклассовых отношений

Межклассовые отношения позволяют соединять классы.

Кратность – количество экземпляров одного класса к другому. Бывает следующих видов:

1. Один к одному (пример: один пользователь и одна роль админа)
2. Один ко многим (пример: один пользователь отыгрывает множество ролей)
3. Многие к многим (пример: множеству юзеров доступно множество ролей)

### Композиция

Композиция – вид зависимости, при которой один класс (класс 1 – класс-часть) является частью другого класса (класс 2– класс-композитор). То есть класс2 содержит в себе класс1. При удалении объекта класс 2, объект класса1 также удаляется.

Объекты классов-частей создаются в конструкторе класса-композитора и удаляется в деструкторе класса класса-композитора. Класс-композитор не может принимать или возвращать где-либо объекты класса-части. Работа с объектами классов-частей ведется только в методах класса-композитора или его наследнике.

Объекты классов частей живут столько же сколько и класс-композитор. Пример человек и его органы.

### Агрегация

Агрегация – тоже самое что и композиция за исключением некоторых аспектов:

1. Класс-агрегатор может получать объект класса-части
2. При удалении класса-агрегатора, объекты класса-части остаются
3. Может получать в качестве аргумента-метода, объект класса-части
4. Может возвращать объект класса-части из метода
5. (Опционально) При изменении где-то объекта класса-части, объект изменится и в классе-агрегаторе (при условии, что объекты изменяемые)

### Ассоциация

Ассоциация – отношение между классами, при котором объект класса-части подается в метод класс-ассоциатор и над ним производятся определенные действия.

Класс-ассоциатор не имеет поля типа класса-части и не может возвращать объекты типа класса-части

Очень часто стрелкой ассоциации заменяют такие отношения как Композиция, Агрегация, Использование.

### Использование

Использование – отношение между классами, при котором объект класса-части создается в метод класс-пользователь и над ним производятся определенные действия по после его возвращают из метода.

Класс-пользователь не имеет поля типа класса-части и не может возвращать объекты типа класса-части

### Наследование\Генерализация

Схематическое отношение между родительским и дочерним классом. По факту является тем же самым что и в ООП.

Очень часто стрелкой наследования заменяют такое отношение как Реализация

### Реализация

Реализация – подразумевает отношение интерфейса и объекта. То есть один объект реализует поведение другого. Пример:

Класс Поставщик имеет методы для покупки товара. Для того чтобы сделать реализацию, класс Клиента должен иметь метод-интерфейс, который будет запрашивать информацию у пользователя и отправлять все это в метод поставщика.

Диаграмма объектов

Диаграмма объектов — это структурная диаграмма, которая показывает объекты, их атрибуты и взаимосвязи с другими объектами.

Диаграмма объектов UML является экземпляром диаграммы классов, она показывает снимок подробного состояния системы в определенный момент времени.

Таким образом, можно сказать, что диаграмма объектов является частным случаем диаграммы классов.

### Использование диаграммы объектов

На этапе анализа проекта можно создать диаграмму классов для описания структуры системы, а затем создать набор объектных диаграмм в качестве тестовых примеров для проверки точности и полноты диаграммы классов.

Прежде чем создавать диаграмму классов, можно создать диаграмму объектов, чтобы узнать факты о конкретных элементах модели и их связях или проиллюстрировать конкретные примеры необходимых классификаторов.

### Отношения между объектами

Между объектами отношения такие же как и между классами

## Диаграмма компонентов

Показывает как компоненты соединяются вместе для формирования более крупных компонентов.

Иллюстрирует архитектуру компонентов и зависимости между ними.

## Диаграмма развертывания

Позволяет моделировать физический аспект объектной ориентированной программной системы.

Схема, которая показывает архитектуру системы как развертывания программных артефактов.

Артефакты – конкретные элементы в физическом мире (является результатом процесса разработки).

Программа моделирует конфигурацию времени выполнения в статическом представлении и визуализирует распределение артефактов в приложении.

## Диаграмма пакетов

Показывает пакеты и зависимости между ними

## Диаграмма составной структуры

Является диаграммой компонентов. Используется при моделировании систем на микро уровне.

Изображает отдельные части вместо целых классов. Показывает внутреннюю структуру класса и возможные взаимодействия.

## Диаграмма профилей

Позволяет создавать стереотипы и определять отношение между ними

# Диаграммы поведения

## Диаграмма прецедентов

Описывает функциональные требования системы с точки зрения прецедентов. Позволяет связать то что нам нужно от системы с тем как система удовлетворяет эти потребности

## Диаграмма деятельности

Представляет рабочие процессы, поэтапных действий и действий с поддержкой выбора итераций и параллелизма. Описывает управление системой

## Диаграмма состояний

Тип диаграмм, используемых для описания поведения системы. Отображают разрешенные состояния и переходы. Позволяет визуализировать весь жизненный цикл объектов.

## Диаграмма взаимодействий

### Диаграмма последовательности

Моделирует взаимодействие объектов на основе временной последовательности. Показывает как объекты взаимодействуют с другими.

### Диаграмма коммуникаций

Используется для моделирования динамических претендентов. Показывает взаимодействие объектов.

### Диаграмма обзора взаимодействий

Фокусируется на обзоре потока управления.

### Временная диаграмма

Показывает поведение объектов в период времени